

IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Currently Amended): A data access, replication or communications system comprising:

a terminal including an electronic memory storing a terminal-side executable and a processor provided to execute the terminal-side executable to enable communication therewith independent of a session-based protocol, the terminal-side executable dividing a message into a plurality of packets, each packet having a size corresponding to a transport protocol payload size; and

a server including an electronic memory storing a server-side executable and a processor provided to execute the server-side executable to enable communication therewith independent of a session-based protocol, the server-side executable dividing a message into a plurality of packets, each packet having a size corresponding to a transport protocol payload size,

wherein [[,]] the terminal-side executable and the server-side executable exchange messages using a message queuing system over a network and cooperatively function as a client of a second server and the server-side executable uses data stored on the server to complete an incomplete message received from the terminal.

Claim 2 (Previously Presented): The system of Claim 1 wherein the message queuing system is message oriented middleware.

Claim 3 (Previously Presented): The system of Claim 1 wherein the terminal-side executable insulates a terminal program from being affected if a connection over the network

is broken by queuing messages in readiness for the connection to be re-established, enabling the terminal program to proceed to another task.

Claim 4 (Previously Presented): The system of Claim 1 wherein the server-side executable insulates a server program from being affected if a connection over the network is broken by queuing messages in readiness for the connection to be re-established, enabling the server program to proceed to another task.

Claim 5 (Previously Presented): The system of Claim 1 wherein each message that is queued defines part or all of an event, the event describing a change to data stored at either the terminal or server in enough detail to enable data replication to take place without a need for a synchronization engine, data replication being achieved by sending events rather than a complete dataset (or sub-sets of a dataset) of stored data for synchronization.

Claim 6 (Previously Presented): The system of Claim 5 wherein the terminal-side executable can create and queue events, enabling the terminal-side executable to proceed to another task, even if a network connection is broken, the events being queued in one of the terminal-side executable and a message queuing system.

Claim 7 (Previously Presented): The system of Claim 5 wherein the server-side executable can create and queue events, enabling the server-side executable to proceed to another task, even if a network connection is broken, the events being queued in one of the server-side executable and a message queuing system.

Claim 8 (Previously Presented): The system of Claim 6 wherein the queued events persist in non-volatile memory when the terminal is switched off.

Claim 9 (Previously Presented): The system of Claim 7 wherein queued events persist in non-volatile memory when the server is switched off.

Claim 10 (Previously Presented): The system of Claim 1 wherein the terminal-side executable and the server-side executable cooperatively function as middleware between a terminal program running on a wireless terminal and a server program running on the server.

Claim 11 (Previously Presented): The system of Claim 6 wherein messages queued on the terminal side are references to data stored on the server.

Claim 12 (Previously Presented): The system of Claim 10 wherein a terminal-side component of the message queuing system insulates the terminal program from being affected if a connection over the network is re-established by automatically causing a next message in a terminal-side queue to be sent.

Claim 13 (Previously Presented): The system of Claim 10 wherein a server-side component of the message queuing system insulates the server program from being affected if a connection over the network is re-established by automatically causing a next message in a server-side queue to be sent.

Claim 14 (Previously Presented): The system of Claim 1 wherein the terminal-side executable processes events from a terminal program, which is an e-mail or PIM program.

Claim 15 (Previously Presented): The system of Claim 1 wherein the server-side executable processes events from a server program, which is a mail server program.

Claim 16 (Previously Presented): The system of Claim 1 wherein the terminal is a wireless terminal such as a mobile telephone or smartphone.

Claim 17 (Previously Presented): The system of Claim 1 wherein the network is a wireless WAN network such as a GPRS or UMTS network.

Claim 18 (Previously Presented): The system of Claim 1 wherein the server-side executable stores a logon password sent from the terminal-side executable and can use the logon password to access a server program.

Claim 19 (Canceled).

Claim 20 (Previously Presented): The system of Claim 1 wherein the terminal-side executable monitors available memory on the terminal and automatically deletes data stored on the terminal that meets a pre-defined criteria of at least one of age, use and size without affecting a corresponding data stored on the server.

Claim 21 (Previously Presented): The system of Claim 20 wherein a user option to delete data stored on the terminal without affecting the corresponding data stored on the server is displayed at a same level in a menu hierarchy, displayed on the terminal, as an

option to delete data stored on the terminal together with the corresponding data stored on the server.

Claim 22 (Previously Presented): The system of Claim 20 wherein the data is message data and the terminal side executable retains data that allows the message data to be re-supplied from the server if requested by a user.

Claim 23 (Previously Presented): The system of Claim 20 wherein data is not released from memory if the data is marked as unread, open for user viewing or action, or there is a pending action related to the data requesting additional data from the second server.

Claim 24 (Previously Presented): The system of Claim 16 wherein the terminal-side executable enables a document attachment to be sent to the wireless terminal in either an original format in which the document is stored on the server or in another useable format converted from the original format.

Claim 25 (Previously Presented): The system of Claim 1 wherein the terminal-side executable enables a user to select a release option to delete a message stored on the terminal without deleting a corresponding message stored on the server and to select a delete option to delete a message stored on the terminal and the corresponding message on the server, the release and delete options appearing at a same level in a menu hierarchy displayed on the terminal.

Claim 26 (Previously Presented): The system of Claim 1 wherein the client, implemented by the terminal-side and server-side executables, enables a correct routing of

messages addressed to a terminal identified by an ID by mapping that ID to an actual IP address needed to reach the terminal.

Claim 27 (Previously Presented): The system of Claim 26 wherein an address is a dynamic IP address allocated by a NAT box.

Claim 28 (Previously Presented): The system of Claim 27 wherein the client, implemented by the terminal-side and server-side executables, only initiates a message transfer if there exists a valid mapping.

Claim 29 (Previously Presented): The system of Claim 28 wherein a mapping is refreshed whenever a specific kind of, dedicated message having fewer than 20 bytes is received from the terminal.

Claim 30 (Previously Presented): The system of Claim 1 wherein the terminal-side executable allows a server administrator to lock an application on the terminal without affecting other applications on the terminal.

Claim 31 (Previously Presented): The system of Claim 1 wherein the terminal-side executable sends a challenge to any third party identified as attempting a denial of service attack on the terminal, preventing any additional data traffic to the terminal from the denial of service attack.

Claim 32 (Previously Presented): The system of Claim 1 wherein the client, implemented by the terminal-side and server-side executables, includes a distributed application platform that makes calls to a distributed communications platform.

Claim 33 (Previously Presented): The system of Claim 32 wherein the distributed communications platform enables reliable delivery of a message over the network independently of any unreliable transport protocol used.

Claim 34 (Currently Amended): A method of data access, replication or communication comprising the steps of:

running a terminal-side executable on a terminal to enable communication with a server independent of a session-based protocol, the terminal-side executable dividing a message into a plurality of packets, each packet having a size corresponding to a transport protocol payload size;

running a server-side executable on the server to enable communication with a terminal independent of a session-based protocol, the server-side executable dividing a message into a plurality of packets, each packet having a size corresponding to a transport protocol payload size, the terminal-side executable and the server-side executable cooperatively functioning as a client to a second server; and

sending messages between the terminal-side executable and the server-side executable using a message queuing system over a network, the server-side executable using data stored on the server to complete an incomplete message received from the terminal.

Claim 35 (Previously Presented): The method of Claim 34, wherein the message queuing system is message oriented middleware,

the terminal-side executable insulates the terminal from being affected by a broken network connection, enabling the terminal to proceed to another task, the terminal-side executable queuing messages in readiness for the network connection to be re-established, and

the server-side executable insulates the server from being affected by a broken network connection, enabling the server to proceed to another task, the server-side executable queuing messages in readiness for the network connection to be re-established.

Claim 36 (Currently Amended): A terminal comprising:

an electronic memory configured to store a terminal-side executable to enable communication therewith independent of a session-based protocol;

a processor configured to execute the terminal-side executable to

divide a message into a plurality of packets, each packet having a size corresponding to a transport protocol payload size,

insulate the terminal from a broken network connection, enabling the terminal to proceed to another task,

queue messages in readiness for the network connection to be re-established, the messages being at least a part of an event describing a change to data stored on a server and allowing data replication without transmission of an entire dataset, and

automatically send the queued messages upon re-establishment of the network connection via at least one of a wired connection or a wireless connection to the network, messages relating to a same dataset being sent through different transmission channels according to message size.



Claim 37 (Currently Amended): A server comprising:

an electronic memory configured to store a server-side executable to enable communication therewith independent of a session-based protocol;

a processor configured to execute the server-side executable to

divide a message into a plurality of packets, each packet having a size corresponding to a transport protocol payload size,

insulate the server from a broken network connection, enabling the server to proceed to another task,

queue messages in readiness for the network connection to be re-established, the messages being at least a part of an event describing a change to data stored on a terminal and allowing data replication without transmission of an entire dataset, [[and]]

automatically sending the queued messages upon re-establishment of the network connection via at least one of a wired connection or a wireless connection to the network, and

using data stored on the server to complete incomplete messages received from the terminal.